

Object REXX for Windows



REXX Mathematical Functions (RxMath)

Version 2.1

Note!

Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 13.

First Edition, March 2001

This edition applies to Version 2.1 of IBM® Object REXX for Windows® Interpreter Edition (5639-M69) and Development Edition (5639-M68), and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

© Copyright International Business Machines Corporation 2001. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Introduction	1	RxCalcTanH()	6
Installation and Removal	1	RxCalcPower()	7
Error Handling	1	RxCalcSin().	7
Chapter 2. Functions	3	RxCalcCos()	8
MathLoadFuncs()	3	RxCalcTan()	8
MathDropFuncs()	4	RxCalcCotan().	9
RxCalcSqrt()	4	RxCalcPi()	9
RxCalcExp()	4	RxCalcArcSin()	10
RxCalcLog()	5	RxCalcArcCos()	10
RxCalcLog10().	5	RxCalcArcTan()	11
RxCalcSinH()	5	Appendix. Notices	13
RxCalcCosH()	6	Trademarks	14

Chapter 1. Introduction

RxMath is a REXX utility package that enables you to use enhanced mathematical functions.

The function names in the REXX Mathematical Functions package are similar to the names of their corresponding mathematical functions.

The precision of calculation depends on:

- The value specified when the command is issued
- The numeric digits settings of the calling REXX activity

Note: Precision is limited to 16 digits.

Installation and Removal

The REXX Mathematical Functions package is contained in the file **rxmath.dll**. This dynamic link library (DLL) must be placed in a directory listed in your PATH. To get access to the functions in the REXX Mathematical Functions package, execute the following REXX code:

```
call RxFuncAdd"MathLoadFuncs", "rxmath", "MathLoadFuncs"  
call MathLoadFuncs
```

To unload the DLL, call the **MathDropFuncs** function and then exit all CMD.EXE shells. After you have exited from all command shells, the DLL is dropped by Windows and can be deleted or replaced.

Error Handling

Error 40 (**Incorrect call to routine**) is raised if either the wrong number of arguments or incorrect data is passed to a function.

If a mathematical function fails, the REXX Mathematical Functions package returns ERROR.

Where an error occurs, the variable **MATHERRNO** is set with additional information that further defines the source of the error.

Chapter 2. Functions

Most function names in the REXX Mathematical Functions package are similar to the names of their corresponding mathematical functions.

- MathLoadFuncs()
- MathDropFuncs()
- RxCalcSqrt()
- RxCalcExp()
- RxCalcLog()
- RxCalcLog10()
- RxCalcSinH()
- RxCalcCosH()
- RxCalcTanH()
- RxCalcPower()
- RxCalcSin()
- RxCalcCos()
- RxCalcTan()
- RxCalcCotan()
- RxCalcPi()
- RxCalcArcSin()
- RxCalcArcCos()
- RxCalcArcTan()

MathLoadFuncs()

►►—MathLoadFuncs()—◄◄

Loads all functions in the REXX Mathematical Functions package.

If you supply any parameters, they will be used only to deliver copyright information.

MathDropFuncs()

►►MathDropFuncs()◄◄

Drops all functions in the REXX Mathematical Functions package.

RxCalcSqrt()

►►RxCalcSqrt(*number* [,*precision*])◄◄

Returns the absolute value of the square root of *number*.

Parameters:

number

The number whose square root you wish to calculate.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcExp()

►►RxCalcExp(*number* [,*precision*])◄◄

Returns the exponential function of *number*.

Parameters:

number

The number for which you wish to calculate the exponential function.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcLog()

►► RxCalcLog(*number* [,*precision*])

Returns the natural logarithm (base e) of *number*.

Parameters:

number

The number for which you wish to calculate the natural logarithm.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcLog10()

►► RxCalcLog10(*number* [,*precision*])

Returns the base 10 logarithm of *number*.

Parameters:

number

The number for which you wish to calculate the base 10 logarithm.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcSinH()

►► RxCalcSinH(*number* [,*precision*])

Returns the hyperbolic sine of *number*, expressed in radians.

Parameters:

number

The number for which you wish to calculate the hyperbolic sine.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcCosh()

► RxCalcCosh(*number* [, *precision*]) ◀

Returns the hyperbolic cosine of *number*, expressed in radians.

Parameters:

number

The number for which you wish to calculate the hyperbolic cosine.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcTanH()

► RxCalcTanH(*number* [, *precision*]) ◀

Returns the hyperbolic tangent of *number*, expressed in radians.

Parameters:

number

The number for which you wish to calculate the hyperbolic tangent.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcPower()

►► RxCalcPower(*number1*, *number2* [, *precision*])

Returns the value of mathematical expression *number1* raised to the power of exponent *number2*.

Parameters:

number1

The mathematical expression to be raised to the power of exponent *number2*.

number2

The exponent to which *number1* is to be raised.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcSin()

►► RxCalcSin(*number* [, *precision*] [,

D
R
G

])

Returns the sine value for *number*, where *number* is the angle size, expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The angle size, expressed in degree [D], radian [R], or grade [G] units.

precision

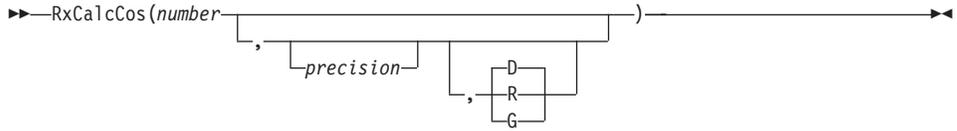
The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D

Indicates that the angle size is expressed in degrees. This is the default.

- R Indicates that the angle size is expressed in radians.
- G Indicates that the angle size is expressed in grades.

RxCalcCos()



Returns the cosine value for *number*, where *number* is the angle size, expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The angle size, expressed in degree [D], radian [R], or grade [G] units.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D

Indicates that the angle size is expressed in degrees. This is the default.

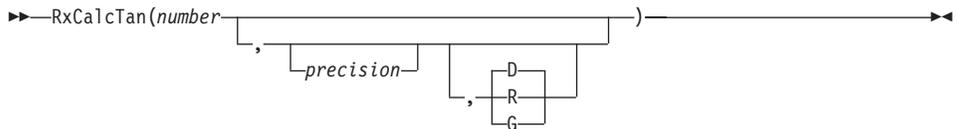
R

Indicates that the angle size is expressed in radians.

G

Indicates that the angle size is expressed in grades.

RxCalcTan()



Returns the tangent value for *number*, where *number* is the angle size, expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The angle size, expressed in degree [D], radian [R], or grade [G] units.

precision

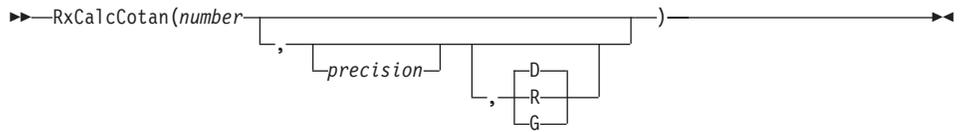
The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D Indicates that the angle size is expressed in degrees. This is the default.

R Indicates that the angle size is expressed in radians.

G Indicates that the angle size is expressed in grades.

RxCalcCotan()



Returns the cotangent value for *number*, where *number* is the angle size, expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The angle size, expressed in degree [D], radian [R], or grade [G] units.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D Indicates that the angle size is expressed in degrees. This is the default.

R Indicates that the angle size is expressed in radians.

G Indicates that the angle size is expressed in grades.

RxCalcPi()



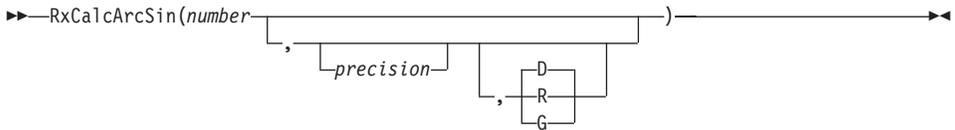
Returns the value of pi.

Parameter:

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

RxCalcArcSin()



Returns the arcsine of *number*, where the result can be expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The number for which the arcsine is to be calculated.

precision

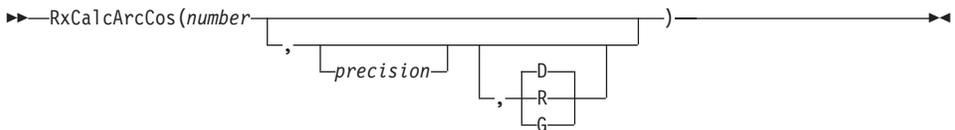
The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D Indicates that the result is expressed in degrees. This is the default.

R Indicates that the result is expressed in radians.

G Indicates that the result is expressed in grades.

RxCalcArcCos()



Returns the arccosine of *number*, where the result can be expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The number for which the arccosine is to be calculated.

precision

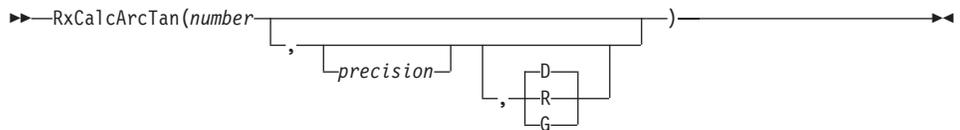
The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D Indicates that the result is expressed in degrees. This is the default.

R Indicates that the result is expressed in radians.

G Indicates that the result is expressed in grades.

RxCalcArcTan()



Returns the arctangent of *number*, where the result can be expressed in degree [D], radian [R], or grade [G] units.

Parameters:

number

The number for which the arctangent is to be calculated.

precision

The precision of the calculation. If a value is not specified, the precision of the calling REXX activity is used. If the precision exceeds 16 digits, the call will fail.

D Indicates that the result is expressed in degrees. This is the default.

R Indicates that the result is expressed in radians.

G Indicates that the result is expressed in grades.

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will

be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland
Informationssysteme GmbH
Department 3982
Pascalstrasse 100
70569 Stuttgart
Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following term is a trademark of the IBM Corporation in the United States, other countries, or both:

IBM

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.